A Turbo-Inspired Iterative Approach for Correspondence Problems of Image Features

Ala Aboudib Vincent Gripon Gilles Coppin

Télécom Bretagne
Technopôle Brest-Iroise, CS 83818 - 29238 Brest cedex 3, France
Lab-STICC UMR CNRS 6285
{name}.{surname}@telecom-bretagne.eu

***Abstract*—Establishing correspondences between image features is a fundamental problem in many computer vision tasks. It is traditionally viewed as a graph matching problem, and solved using an optimization procedure. In this paper, we propose a new approach to solving the correspondence problem from a coding/decoding perspective. We then present an iterative matching algorithm inspired from the turbo-decoding concept. We provide an experimental evaluation of the proposed method, and show that it performs better than state-of-the-art algorithms in the presence of clutter, thanks to turbo-style decoding.**

I. Introduction

The problem of finding correspondences between features of two images is fundamental to computer vision. Solving this problem would be of particular importance to a variety of vision tasks. This includes object tracking [1], object recognition [2], stereo matching [3] and other tasks.

The basic idea is simple: given two images m and m', where m contains only one object b (the query object), we are interested in finding a possibly deformed instance b' of b in the image m', knowing that m' might contain other objects than the one in question. In order to achieve that, we take two sets of local image features $\mathcal{V}$ and $\mathcal{V}'$ representing m and m', respectively. Then, we search a mapping from $\mathcal{V}$ to $\mathcal{V}'$ that is injective.

While mapping features based on the similarity among their descriptor vectors can give good matches in simple cases, this does not hold in more difficult situations where image m' is cluttered, which is very common in natural scenes.

Considering geometrical consistency between features in addition to their descriptor similarity was suggested as a better way to achieve correct matching. For instance, in early methods such as RANSAC [4] and ICP [5], a solution is accepted only if the matched features in $\mathcal{V}'$ are constrained to some parametric transformation (e.g. epipolar or affine) of their counterparts in $\mathcal{V}$. However, given that non-rigid transformations are very common in natural images, applying these parametric constraints becomes a limitation in such cases.

This work was supported by the European Research Council under the European Union's Seventh Framework Program (FP7/2007-2013) / ERC grant agreement n° 290901.

In order to take both feature similarity and their geometrical proximity into account, including the case of non-rigid deformations, a class of methods were proposed in the last two decades that formulated feature matching as a graph matching (GM) problem; two graphs $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$ are constructed on the sets of features $\mathcal{V}$ and $\mathcal{V}'$ representing the graph nodes. Graph edges in $\mathcal{E}$ and $\mathcal{E}'$ are assigned values of some measure of geometrical proximity between pairs of nodes in $\mathcal{V}$ and $\mathcal{V}'$, respectively. Then we search the sub-graph of $\mathcal{G}'$ that best matches $\mathcal{G}$ in terms of unary feature similarity and pairwise geometric consistency.

This graph matching problem constrained to an injective mapping from $\mathcal{V}$ to $\mathcal{V}'$ is known to be NP-hard. A whole class of methods proposed to approach it as a Quadratic Assignment Problem (QAP) [6]–[8], where an approximate solution can be obtained by optimizing a well-defined objective function. Some of these methods suggested an iterative approach to optimizing this objective function such as the max-pooling matching (MPM) [9], spectral matching (SM) [10], re-weighted random walks (RRWM) [11], balanced graph matching (BGM) [12].

The main contribution of this paper is to propose an iterative matching algorithm, which uses turbo decoding principles as an inspiration [13], and provides a better matching accuracy of features in cluttered images. The turbo concept is used to enforce the injective mapping constraint at each iteration. Actually, the injective mapping constraint from $\mathcal{V}$ to $\mathcal{V}'$ implies two different constraints: (1) A feature $v_i \in \mathcal{V}$ is allowed to match at most one feature in $\mathcal{V}'$ (by the definition of a mapping). (2) A feature $v'_a \in \mathcal{V}'$ is allowed to match at most one feature in $\mathcal{V}'$ (injectivity constraint). Unlike conventional algorithms, we neither relax these constraints nor we enforce them both at the same time. Each iteration of the algorithm we propose enforces one of these constraints at a time. It alternates between them at each iteration until a good match is obtained.

The rest of this paper is organized as follows. In section II we present the mathematical formulation of the matching problem in question, and we explain how it relates to coding theory. After that, in section III, we introduce our iterative matching algorithm inspired from the concept of turbo decoding. An experimental evaluation of the performance of the proposed

978-1-5090-3401-7/16/.00 ©2016 IEEE

algorithm with a comparison with some of the state-of-the-art methods is provided in section IV, along with a discussion of the results. Section V is a conclusion.

II. Problem Statement

A. Formalism

In this paper, we follow the graph matching approach (GM) to the correspondence problem. The objective is to match a query graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, to a sub-graph of $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$. We define an assignment matrix $\mathbf{X} \in \{0,1\}^{nn'}$ as in [10], where $n = |\mathcal{V}|$ and $n' = |\mathcal{V}'|$. Elements of $\mathbf{X}$ are set as follows:

$$\mathbf{X}_{ia} = \begin{cases} 1 & \text{if feature } v_i \text{ matches } v'_a, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

We also use an assignment vector $\mathbf{x}$, which is a column-wise vectorized copy of $\mathbf{X}$. We define a unary affinity function $S_V(v_i, v'_a)$ to measure the similarity between two feature descriptors, and a pairwise affinity function $S_E(e_{ij}, e'_{ab})$ that measures similarity between two edges $e_{ij} \in \mathcal{E}$ and $e'_{ab} \in \mathcal{E}'$. We use these functions to populate a unary affinity vector as $\mathbf{y}_{ja} = S_V(v_i, v'_a)$, and a pairwise affinity matrix $\mathbf{A} \in \mathbb{R}^{nn' \times nn'}$:

$$\mathbf{A}_{ia;jb} = \begin{cases} S_E(e_{ij}, e'_{ab}) & \text{if } i \neq j \text{ and } a \neq b, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

An objective function is defined using the above affinity functions:

$$f(\mathbf{x}) = \sum_{\substack{\mathbf{x}_{ia}=1 \\ \mathbf{x}_{jb}=1}} S_E(e_{ij}, e'_{ab}) + \sum_{\mathbf{x}_{ia}=1} S_V(v_i, v'_a). \quad (3)$$

This is a known quadratic assignment problem (QAP) that can be written in matrix form as:

$$f(\mathbf{x}) = \mathbf{x}^{\mathsf{T}}(\mathbf{A} + diag(\mathbf{y}))\mathbf{x}, \quad (4)$$

where $diag(\mathbf{y})$ is a square matrix that contains zeros everywhere except on its main diagonal where it holds the vector $\mathbf{y}$. The solution to this problem can be expressed as the assignment vector $\mathbf{x}^*$ that maximizes the objective function $f(\mathbf{x})$:

$$\begin{aligned} \widetilde{\mathbf{x}}^* &= \arg\max_{\widetilde{\mathbf{x}}} \widetilde{\mathbf{x}}^{\mathsf{T}}(\mathbf{A} + diag(\mathbf{y}))\widetilde{\mathbf{x}}, & (5) \\ \mathbf{x}^* &= z(\widetilde{\mathbf{x}}^*), & (6) \\ & \quad s.t.\ \ \mathbf{x}^* \in \{0,1\}^{nn'}, \widetilde{\mathbf{x}} \in [0,1]^{nn'}, \\ & \quad \text{and } \mathbf{x}^* \text{ represents an injective mapping} \\ & \quad \text{from } \mathcal{V} \text{ to } \mathcal{V}'. \end{aligned}$$

Notice that the constraint on $\mathbf{x}$ being discrete is relaxed during the optimization process, this relaxed version of the assignment vector is denoted $\widetilde{\mathbf{x}}$. Notice also that the objective function does not enforce the injective mapping constraint from $\mathcal{V}$ to $\mathcal{V}'$ we are seeking. This constraint is usually relaxed during the optimization procedure to reduce the complexity of the problem.

The final continuous assignment vector $\widetilde{\mathbf{x}}^*$ obtained is then discretized in (6) after applying a greedy or a Hungarian algorithm that enforces injective mapping and the discrete-value constraints [9], [10], [14].

The algorithm we propose follows a different procedure; while $\mathbf{x}$ is allowed to be continuous during the process, the injective mapping constraint is not totally relaxed during optimization; they are enforced at each iteration, alternating between the mapping constraint and the injectivity one, until a satisfying solution is obtained.

B. Relation to the Coding Theory

One way to relate the correspondence problem to a coding/decoding procedure is illustrated in figure 1. In this configuration, the query graph $\mathcal{G}$ is treated as the transmitted codeword, and the destination graph $\mathcal{G}'$ as the observation, which is viewed as a corrupted version of $\mathcal{G}$ due to the noisy transmission channel.

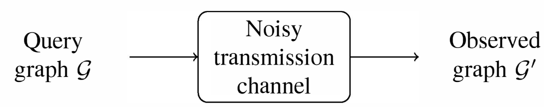

Fig. 1: Feature matching viewed as transmission problem.

The noise in the transmission channel is due to three different factors:

- Spatial deformation of feature locations in $\mathcal{V}'$ compared to their counterparts in $\mathcal{V}$ due to all kinds of rigid and non-rigid object transformations.
- The intrinsic ambiguity of the problem in some cases, where more than one matching solution might be possible. One good example is in the case of matching features having an equilateral triangular configuration in each image, with a pairwise affinity function $S_E(.)$ that only considers relative positions of features. In this case, each feature in $\mathcal{V}$ can match any feature in $\mathcal{V}'$.
- The presence of outliers (clutter) which are features that do not belong to the objects we are trying to match.

However, since we are seeking to find a match among graph nodes rather than to recover the graph $\mathcal{G}$ from the observation, a better way to build the transmission network is to take a ground truth assignment vector $\mathbf{x}$ as the transmitted codeword. The pairwise affinity matrix $\mathbf{A}$ and the unary affinity vector $\mathbf{y}$ are the observed variables as depicted in figure 2. Our objective is then to decode our observations in order to get the vector $\mathbf{x}^*$ belonging to the constrained domain $\{0,1\}^{nn'}$:

We are particularly interested in what happens inside the decoder in figure 2. The most common method in literature is to apply an optimization procedure to find $\widetilde{\mathbf{x}}^*$ as in (5). Then discretization is applied on that vector as in (6). In the next section, we will present our proposed decoding method inspired from the turbo decoding concept, and how it differs from classical methods.

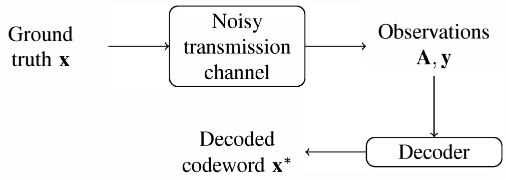

Fig. 2: The matching problem viewed as an error correcting problem of a codeword received through a noisy transmission channel.

III. Methodology

The architecture of the decoding process we propose is depicted in figure 3. Here is a list of all signals manipulated and produced by that process:

- The pairwise affinity matrix $\mathbf{A}$.
- The unary affinity vector $\mathbf{y}$.
- Relaxed assignment vectors $\widetilde{\mathbf{x}} \in [0,1]^{nn'}$. These vectors are called relaxed because they do not respect the injective mapping constraint.
- Semi-relaxed assignment vectors $\bar{\mathbf{x}} \in [0,1]^{nn'}$. They are semi-relaxed because they partially enforce one of the two constraints; injectivity or mapping at each time. These vectors are sparse; most of their elements are set to zero.
- The final assignment vector $\mathbf{x}^*$. The assignment described by this vector respects both the mapping and the injectivity constraints.

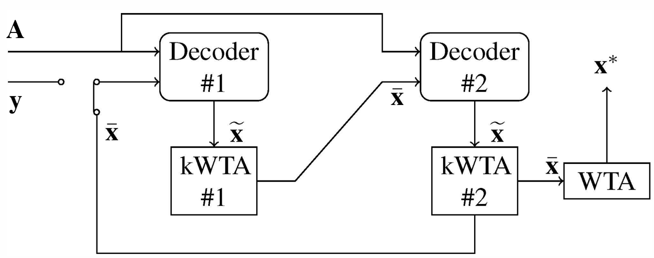

Fig. 3: The architecture of the proposed decoder.

Decoder units

Each decoder unit takes two inputs: the observation $\mathbf{A}$ and either the unary affinity vector $\mathbf{y}$, or a semi-relaxed assignment vector $\bar{\mathbf{x}}$. The vector $\mathbf{y}$ is only taken by the first decoder in the first iteration. In all subsequent iterations, the vector $\bar{\mathbf{x}}$ is used instead. The output of each decoder unit is a relaxed assignment vector $\widetilde{\mathbf{x}}$. This vector is computed as a max-pooled weighted sum of elements in $\mathbf{A}$ as follows:

$$\widetilde{\mathbf{x}}_{ia} \leftarrow \bar{\mathbf{x}}_{ia} \sum_{j \in \mathcal{V}} \max_{b \in \mathcal{V}'} \bar{\mathbf{x}}_{jb} \mathbf{A}_{ia;jb}. \tag{7}$$

This equation is applied by the first decoder. Notice that pooling is applied on elements in $\mathcal{V}'$ as in [9]. The second decoder applies max-pooling on elements in $\mathcal{V}$:

$$\widetilde{\mathbf{x}}_{ia} \leftarrow \bar{\mathbf{x}}_{ia} \sum_{b \in \mathcal{V}'} \max_{j \in \mathcal{V}} \bar{\mathbf{x}}_{jb} \mathbf{A}_{ia;jb}. \tag{8}$$

The operation applied by each of these decoders is akin to the power method used in spectral matching (SM) [10] to find the first eigen vector of matrix $\mathbf{A}$. Max-pooling is added to discard irrelevant details while preserving necessary information as in [9].

kWTA units

Each k-winner take all (kWTA) unit takes a relaxed assignment vector $\widetilde{\mathbf{x}}$ as its input, and produces a semi-relaxed assignment vector $\bar{\mathbf{x}}$ as an output. The first kWTA unit is only concerned about the mapping constraint. It 'encourages' the vector $\widetilde{\mathbf{x}}$ to respect that constraint without strictly enforcing it. In other words, it reduces the number of matches in $v_a \in \mathcal{V}'$ that a single feature $v_i \in \mathcal{V}$ can take. This is done by applying a kWTA operation as follows:

$$\begin{aligned} \widetilde{\mathbf{x}}_{ia} &\longleftarrow \frac{\widetilde{\mathbf{x}}_{ia}}{\max_{a \in \mathcal{V}'} \widetilde{\mathbf{x}}_{ia}}, && (9) \\ \bar{\mathbf{x}}_{ia} &\leftarrow \widetilde{\mathbf{x}}_{ia} h(\widetilde{\mathbf{x}}_{ia} - \tau), && (10) \\ &\forall i \in \mathcal{V}, a \in \mathcal{V}', \end{aligned}$$

where $h(.)$ is the unit step function and τ is the kWTA activation threshold.

The second kWTA units applies a similar operation for the injectivity constraint to reduce the number of features in $\mathcal{V}$ mapped to a single feature in $\mathcal{V}'$:

$$\begin{aligned} \widetilde{\mathbf{x}}_{ia} &\longleftarrow \frac{\widetilde{\mathbf{x}}_{ia}}{\max_{i \in \mathcal{V}} \widetilde{\mathbf{x}}_{ia}}, && (11) \\ \bar{\mathbf{x}}_{ia} &\leftarrow \widetilde{\mathbf{x}}_{ia} h(\widetilde{\mathbf{x}}_{ia} - \tau), && (12) \\ &\forall i \in \mathcal{V}, a \in \mathcal{V}'. \end{aligned}$$

Notice that the $\max$ function in (11) is applied across elements of $\mathcal{V}$, while in (9), it is applied accross elements of $\mathcal{V}'$. The output $\bar{\mathbf{x}}$ of the second kWTA unit is used in the next iteration as an input to the first decoder unit. This iterative process stops when the vector $\bar{\mathbf{x}}$ converges. At this point, the vector $\bar{\mathbf{x}}$ will be used as an input to the WTA unit in order to compute the final output $\mathbf{x}^*$. However, since a theoretical guarantee for convergence is yet to be proved, we typically fix a maximum number of allowed iterations beyond which the process terminates.

WTA unit

The winner-takes-all (WTA) unit takes a semi-relaxed assignment vector $\bar{\mathbf{x}}$ as an input and produces the final assignment vector $\mathbf{x}^* \in \{0,1\}^{nn'}$, which respects the injectivity mapping constraint. The first step is to zero all values in $\bar{\mathbf{x}}$ that do not equal one, which is the maximal values in $\bar{\mathbf{x}}$:

$$\begin{aligned} \mathbf{x}^*_{ia} &\leftarrow \delta_1^{\bar{\mathbf{x}}_{ia}}, && (13) \\ &\forall i \in \mathcal{V}, a \in \mathcal{V}', \end{aligned}$$

where δ is the Kronecker delta. After that, each non-zero value $\bar{\mathbf{x}}_{ia}$ is set to zero if there exists at least one non-zero value of the form $\bar{\mathbf{x}}_{ik}$ or $\bar{\mathbf{x}}_{ka}$ different from $\bar{\mathbf{x}}_{ia}$. By applying this procedure, the resulting assignment vector $\bar{\mathbf{x}}$ is guaranteed to respect the injective mapping constraint.

IV. Results and Discussion

We use experimental evaluation to assess the performance of our algorithm. A typical evaluation method used in feature matching literature is accomplished using synthetic datasets. We create two sets P and P' containing points in $\mathbb{R}^2$. Graphs $\mathcal{G}$ and $\mathcal{G}'$ are created using P and P', respectively. Each set contains two types of points: inliers and outliers. Inliers are points representing features that we are seeking to match. Outliers, on the other hand, are points that represent features that describe clutter or noise that we wish to ignore during the matching process.

We randomly generate n_{in} inliers with coordinates sampled uniformly from the interval $[-1, +1]$, and we add them to P. We then add a Gaussian noise $\mathcal{N}(0, \sigma^2)$ to each of these inliers before adding them to the set P'. After that, we add n_{out} outliers generated from the same distribution as the inliers to each of P and P'.

The unary affinity function is considered to be always constant $S_V(p_i, p'_a) = 1$, while the pairwise affinity function is defined as follows:

$$S_E(e_{ij}, e'_{ab}) = \exp(-\big|\|p_i - p_j\| - \|p'_a - p'_b\|\big|). \tag{14}$$

Using a constant $S_V(.)$ represents a difficult case where matching depends only on the geomterical consistency of features. We set the kWTA threshold $\tau = 0.98$, which we found to give the best accuracy. Convergence of the algorithm is attained after $5 - 10$ iterations in most cases. Therefore, the maximum number of allowed iterations is set to 10.

We first evaluate the performance of the proposed model in the presence of outliers. We refer to the proposed model by the term 'turbo matcher' to emphasize the turbo-style decoding used. In all of our experiments, performance is measured in terms of accuracy, which is the percentage of the number of correct matches to the total number of inliers.

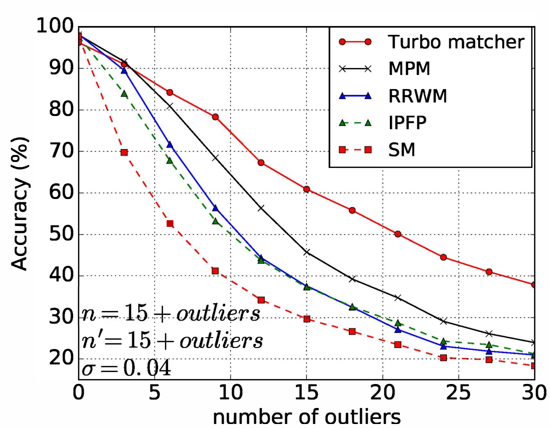

Fig. 4: A comparison among models' accuracy in the presence of outliers. The number of outliers is varied for a fixed value of σ. The same number of outliers shown on the horizontal axis is added to both sets $\mathcal{V}$ and $\mathcal{V}'$.

In the first experiment, we fix the number of inliers to $n_{in} = 15$ in both sets $\mathcal{V}$ and $\mathcal{V}'$, the standard deviation of the Gaussian noise to $\sigma = 0.04$, and we vary the number of outliers in both sets as shown in figure 4. The accuracy of the proposed model is then compared to some state-of-the-art matching algorithms including MPM [9], RRWM [11], IPFP [14] and SM [10]. We notice that the accuracy of the turbo matcher surpasses state-of-the-art by a significant margin, even when the number of outliers is twice the number of inliers. This robustness to outliers is a very interesting property since outliers in the form of clutter and noise are omnipresent in natural images.

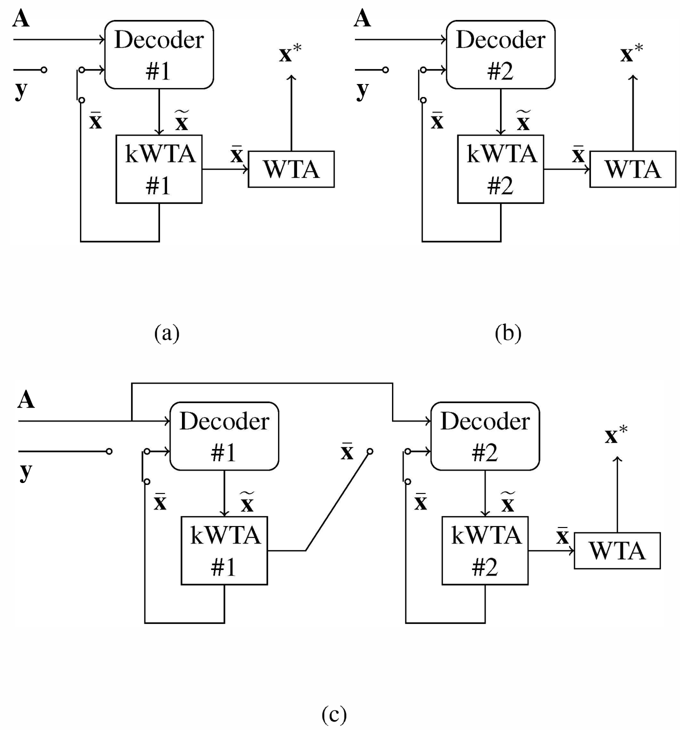

Fig. 5: Other possible configurations of decoder and kWTA units. In (a) and (b), only one decoder and one kWTA units are used. In (c), both decoder units and kWTA units are used, but no alternation between constraints is involved.

In a second experiment, we evaluate the performance gain offered by the alternating double-decoder scheme of figure 3. In other words, we try to answer the question of whether alternating between decoders is behind the performance gain we observe, or there exists other configurations that give a comparable performance. In order to do that, we compare the turbo matcher with three alternate configurations illustrated in figure 5: (1) in the absence of decoder#1 and kWTA#1, (2) in the absence of decoder#2 and kWTA#2, (3) without alternation between the two constraints. In the later test case, two separate iterative phases are run consecutively. The first one includes decoder#1 and kWTA#1 which 'encourages' the mapping constraint. The second phase includes only decoder#2 and kWTA#2, and 'encourages' the injectivity constraint.

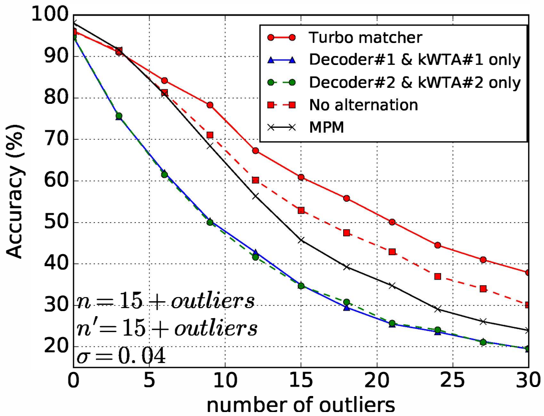

Fig. 6: Performance gain offered by turbo-style decoding. We show how turbo-style alternation between decoders gives a better accuracy than using only one decoder or using both decoders consecutively rather than in an alternating fashion.

Figure 6 shows that enforcing both the mapping and the injectivity constraints, whether in an alternating or a non-alternating fashion gives a better accuracy than using only one decoder unit with its associated kWTA unit. However, turbo-style alternating between decoder units gives a better accuracy than enforcing constraints separately without alternation.

The final experiment consists in fixing the number of inliers to 30 with no outliers. The parameter σ is then varied. The accuracy of the turbo matcher is evaluated, and compared to state-of-the-art for each value of σ. We notice in figure 7 that the turbo matcher gives a rather modest accuracy in this case outperformed by both IPFP and RRWM. However, as stated in [9]: while a better performance in the absence of outliers might be interesting in some situations, it is not a sufficient property from a practical point of view, since outliers are always present in natural images. In such cases, robustness to outliers is an indispensable property for matching algorithms to be equipped with.

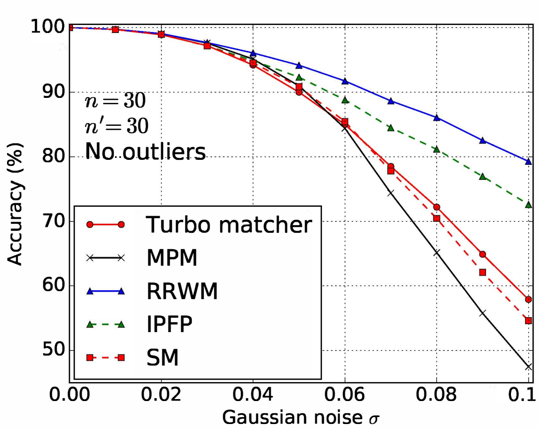

Fig. 7: A performance comparison in the absence of outliers. The standard deviation σ of the Gaussian noise is varied, and accuracy is evaluated at each step.

V. Conclusion and Future Work

In this paper, we presented a new way to view the feature correspondence problem as a coding/decoding procedure inspired by turbo principles. We showed that by using an iterative approach that enforces each of the problem constraints separately but in an alternating fashion, we obtain a better performance in the presence of outliers. In future work, it would be interesting to suggest a theoretical analysis of the convergence of the proposed iterative algorithm. It would also be interesting to provide a performance evaluation of the turbo matcher on natural images, and to propose a more elaborate way of enforcing the injective mapping constraint in the WTA unit.

References

[1] H. Jiang, S. X. Yu, and D. R. Martin, "Linear scale and rotation invariant matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1339–1355, July 2011.

[2] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Tenth IEEE International Conference on Computer Vision (ICCV), 2005.*, vol. 2, Oct 2005, pp. 1458–1465 Vol. 2.

[3] T. Tuytelaars and L. V. Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *British Machine Vision Conference (BMVC), 2000.*, 2000, pp. 412–425.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[5] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[6] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.*, June 2008, pp. 1–8.

[7] O. Duchenne, F. Bach, I. S. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2383–2395, Dec 2011.

[8] F. Zhou and F. D. la Torre, "Factorized graph matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*, June 2012, pp. 127–134.

[9] M. Cho, J. Sun, O. Duchenne, and J. Ponce, "Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014*, June 2014, pp. 2091–2098.

[10] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Tenth IEEE International Conference on Computer Vision (ICCV), 2005.*, vol. 2, Oct 2005, pp. 1482–1489 Vol. 2.

[11] M. Cho, J. Lee, and K. M. Lee, *11th European Conference on Computer Vision (ECCV), 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ch. Reweighted Random Walks for Graph Matching, pp. 492–505.

[12] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 313–320. [Online]. Available: http://papers.nips.cc/paper/2960-balanced-graph-matching.pdf

[13] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, Oct 1996.

[14] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference," in *Advances in neural information processing systems*, 2009, pp. 1114–1122.